1. 定位

（1）元素可以自由在一 个盒子内移动位置，并且压住其他盒子；当滚动屏幕的时候，有些盒子固定不动；将盒子定在某一个位置，只是按照定位的方式移动盒子。定位=定位模式+边偏移,定位模式：制定一个元素在文档中的定位方式。边偏移：决定了元素的最终位置。

（2）定位模式

通过css的position属性来设置，其值可以分为四个：静态定位（static）可以认为就是标准流，相对定位（relative）绝对定位（absolute）固定定位（fixed）

（3）边偏移

有top，bottom，left，right四个属性

1. 相对定位

元素移动的时候，是相对于他原来的位置来说（自恋型）；虽然这个盒子走了，但是他的位置依然存在，不脱标，最典型的应用是给绝对定位当爹的

语法:选择器{ position：relative；}

1. 绝对定位

元素移动的时候，是相对于祖先元素（拼爹型）；如果没有祖先元素或者没有祖先元素定位，则以浏览器为准定位（document文档）；

语法：选择器{position：absolute；}

如果父元素有定位（相对，绝对，固定），则最近一级的带有定位祖先元素为参考点移动位置；绝对定位不在占有原先的位置（脱标）

1. 子绝父相

子级是绝对定位，父级是相对定位。

子级绝对定位，不会占有位置，可以放到父盒子里面任何一个地方，不会影响其他兄弟盒子；父盒子需要加定位限制子盒子再付盒子内显示；父盒子布局时，需要占有位置，因此父亲只能是相对定位。

总结:因为父级需要占有位置，因此是相对定位，子盒子不要占有位置，则是绝对定位。

1. 固定定位fixed(重要)

元素固定于浏览器可视区的位置，主要使用场景：可以在浏览器页面滚动时元素的位置不会改变

选择器{position：fixed;}

以浏览器的可视窗口为参照点移动元素，跟父元素没有任何关系，不随滚动条滚动，固定定位不占有原来位置（脱标）

小技巧：固定在版心右侧位置（小算法：让固定定位的盒子left%，走到浏览器可视区，也可看做是版心的一半位置；margin-left版心宽度一半距离，多走版心宽度的一般位置）

1. 粘性定位stiky（了解）

相对定位和固定定位的结合，刚开始随之滚动，后来就固定位置

以浏览器的可视窗口为参照点移动元素（固定定位特点）

粘性定位占有原先的位置上（相对定位的特点）

必须添加top、left、right、bottom其中一个才有效

1. 定位叠放顺序 z-index

在使用定位布局时，可能会出现盒子叠加的情况，此时可以使用z-index来控制盒子的前后次序（z轴）语法：选择器{z-index：1}，数值可以是正整数，负整数或0，数值越大 盒子越靠上；如果属性值相同，则后来者居上；数字后面不能加单位；只有定位的盒子才有z-index属性

1. 定位的拓展
2. 绝对定位的盒子居中

加了绝对定位的盒子不能通过加margin：auto来水平居中，但是可以通过算法来实现水平居中（left走父级宽度的一半，margin 负值往左边走盒子宽度一半）和垂直居中（top走父级一半，margin负值往上走盒子宽度一半）

1. 定位的特殊性

绝对定位和固定定位也和浮动类似，行内元素添加绝对定位或相对定位，可以直接设置宽度和高度；块级元素添加绝对定位或相对定位，不设置默认宽度，即是字体高度和宽度；

1. 浮动元素，绝对定位（固定定位），元素都不会触发外边距合并的问题
2. 绝对定位会完全压住盒子

浮动元素不同，只会压住他下面标准流的盒子，但是不回压住下面标准流盒子里面的文字；但是绝对定位会压住线面标准流的所有内容；浮动产生的目的就是做文字环绕效果

1. 网页布局总结

通过盒子模型，清楚知道大部分html标签是一个盒子。

通过css浮动和定位让每个盒子排列成为网页。

一个完整的网页，是标准流，浮动，定位一起布局完成的。

垂直的块级盒子就用标准流；浮动让块级盒子左右一行显示；定位让盒子前后叠加，让元素在某个盒子内移动

1. 元素的显示与隐藏

让一个元素在页面中隐藏或者显示出来

1. display显示隐藏

设置一个元素应如何显示，none隐藏对象，block除了转换为块级元素，同时还有显示元素的意思，隐藏元素后，不再占用位置，搭配js可以做很多特效

1. disibility显示隐藏

用于指定一个元素是可见还是隐藏，visible元素可见，hidden元素隐藏，隐藏后继续站由原来的位置。

1. overflow溢出显示隐藏

指定了如果内容溢出一个元素的框（超过其指定高度及宽度）时，会发生什么。

Visible可见，overflow隐藏，scroll溢出部分显示滚动条（auto溢出时显示滚动条，不溢出不显示）

1. 精灵技术

浏览器发送请求，服务器接收请求，返回请求页面。当页面图像过多时，服务器就会频繁地接收和发送请求图片，造成服务器压力过大，这将大大降低页面的加载速度，为了减少服务器接收和发送请求的次数，提高页面的加载速度，出现css精灵技术。

核心原理：将网页的一些小背景图像整合到一张大图中，这样服务器只需要一次请求就可以了。

1. 精灵图（sprites）使用

把多个小背景整合到一张大图中，移动背景图片的位置，此时可以用background-position，移动距离就是这个目标图片的x和y坐标，注意网页中的坐标有所不同，